

## Claims

I claim:

1. A system for logging an event, comprising:
  - a computer system including a CPU and memory;
  - a server-side logging component comprised in the memory of the computer system;
  - a means for persistently storing information;
  - a first client-side logging component comprised in the memory of the computer system;
  - a first executable module, wherein the first executable module is executable to call the first client-side logging component in order to log an event, wherein the event comprises an event level;
    - wherein the first executable module calling the first client-side logging component comprises the first executable module passing information specifying the event level to the first client-side logging component,
    - wherein the first client-side logging component calling the server-side logging component comprises the first client-side logging component passing information specifying the event level to the server-side logging component;
    - wherein the server-side logging component persistently logging the event comprises the server-side logging component persistently logging the event level;
    - wherein, in response to said first executable module calling the first client-side logging component in order to log an event, the first client-side logging component is executable to call the server-side logging component in order to log the event;
    - wherein, in response to said first client-side logging component calling the server side logging component in order to log the event, the server-side logging component is executable to utilize the means for persistently storing information in order to persistently log the event.

2. The system of claim 1,  
wherein the first client-side logging component maintains logging criteria information specifying which levels of events should be logged;  
wherein, in response to said first executable module calling the first client-side logging component, the first client-side logging component is executable to compare the information specifying the event level to the logging criteria information, in order to determine whether the event should be logged;  
wherein, if the first client-side logging component determines that the event should not be logged, the client-side logging component does not perform said calling the server-side logging component.
3. The system of claim 1,  
wherein the event is associated with a first event category;  
wherein said first executable module calling the first client-side logging component comprises the first executable module passing information specifying the first event category to the first client-side logging component;  
wherein said first client-side logging component calling the server-side logging component comprises the first client-side logging component passing information specifying the first event category to the server-side logging component;  
wherein said server-side logging component persistently logging the event comprises the server-side logging component persistently logging the first event category.
4. The system of claim 3,  
wherein the first client-side logging component maintains logging criteria information specifying which categories of events should be logged;  
wherein, in response to said first executable module calling the first client-side logging component, the first client-side logging component is executable to compare the information specifying the first event category to the logging criteria information, in order to determine whether the event should be logged;  
wherein, if the first client-side logging component determines that the event should not be logged, the client-side logging component does not perform said calling the server-side logging component.

5. The system of claim 4,  
wherein the event is also associated with a second event category;  
wherein said first executable module calling the first client-side logging component further comprises the first executable module passing information specifying the second event category to the first client-side logging component;  
wherein, in response to said first executable module calling the first client-side logging component, the first client-side logging component is executable to compare the information specifying the second event category to the logging criteria information, in order to determine whether the event should be logged.
6. The system of claim 1,  
wherein the first client-side logging component executes in-process with the first executable module.
7. The system of claim 1, further comprising:  
a second executable module, wherein the second executable module is executable to call the first client-side logging component in order to log an event;  
wherein the first executable module and the second executable module and the first client-side logging component run in the same process.
8. The system of claim 7,  
wherein the first client-side logging component executes as an in-process COM object.
9. The system of claim 1, further comprising:  
a second client-side logging component comprised in the memory of the computer system;  
a second executable module, wherein the second executable module is executable to call the second client-side logging component in order to log an event;  
wherein the first executable module and the first client-side logging component run in a first process;  
wherein the second executable module and the second client-side logging component run in a second process.

10. A system for logging an event, comprising:  
a computer system including a CPU and memory;  
a server-side logging component comprised in the memory of the computer system;  
a means for persistently storing information;  
a first client-side logging component comprised in the memory of the computer system;

a first executable module, wherein the first executable module is executable to call the first client-side logging component in order to log an event;

wherein, in response to said first executable module calling the first client-side logging component in order to log an event, the first client-side logging component is executable to store the event in a queue and return execution control to the first executable module prior to evaluating the event;

wherein said first client-side logging component calling the server-side logging component comprises the first client-side logging component asynchronously retrieving the event from the queue and then calling the server-side logging component;

wherein, in response to said first executable module calling the first client-side logging component in order to log an event, the first client-side logging component is executable to call the server-side logging component in order to log the event;

wherein, in response to said first client-side logging component calling the server-side logging component in order to log the event, the server-side logging component is executable to utilize the means for persistently storing information in order to persistently log the event.

11. The system of claim 10,  
wherein said first client-side logging component asynchronously retrieving the event from the queue and then calling the server-side logging component is performed by an event queue manager thread.

12. The system of claim 1, further comprising:  
a logging administration tool executable to configure the server-side logging component with event-logging criteria information, wherein the event-logging criteria information includes information specifying which levels of events should be logged;

wherein the first client-side logging component maintains logging criteria information specifying which levels of events should be logged;

wherein the server-side logging component is executable to propagate event-logging criteria information to the first client-side logging component in response to said logging administration tool configuring the server-side logging component with event-logging criteria information.

13. The system of claim 12,

wherein the first client-side logging component is executable to update the logging criteria information that it maintains, in response to said server-side logging component propagating the event-logging criteria information to the first client-side logging component.

14. The system of claim 13,

wherein the server-side logging component is a COM component;

wherein the first client-side logging component is a COM component;

wherein said server-side logging component propagating the event-logging criteria information to the first client-side logging component is performed by using COM mechanisms for sending a COM event from one COM component to another COM component.

15. The system of claim 1, the system further comprising:

a logging administration tool executable to configure the server-side logging component with event-logging criteria information, wherein the event-logging criteria information includes information specifying which categories of events should be logged;

wherein the first client-side logging component maintains logging criteria information specifying which categories of events should be logged;

wherein the server-side logging component is executable to propagate event-logging criteria information to the first client-side logging component in response to said logging administration tool configuring the server-side logging component with event-logging criteria information;

wherein the first client-side logging component is executable to update the logging criteria information that it maintains, in response to said server-side logging component propagating the event-logging criteria information to the first client-side logging component.

16. The system of claim 1,  
wherein, in response to said first executable module calling the first client-side logging component in order to log an event, the first client-side logging component is executable to timestamp the event;  
wherein said client-side logging component calling the server-side logging component comprises the client-side logging component passing timestamp information for the event to the server-side logging component;  
wherein said server-side logging component utilizing the means for persistently storing information in order to persistently log the event comprises the server-side logging component logging the timestamp information.
17. The system of claim 16,  
wherein the server-side logging component is executable to utilize event timestamp information in order to determine the order in which events should be logged.
18. The system of claim 1,  
wherein the means for persistently storing information comprises an element from the group consisting of:  
a file and a database.
19. A system for logging an event, comprising:  
a first computer system including a CPU and memory;  
a second computer system including a CPU and memory;  
a server-side logging component comprised in the memory of the first computer system;  
a means for persistently storing information;  
a first client-side logging component comprised in the memory of the second computer system;  
a first executable module running on the second computer system, wherein the first executable module is executable to call the first client-side logging component in order to log a first event;  
wherein, in response to said first executable module calling the first client-side logging component in order to log a first event, the first client-side logging component is executable to

call the server-side logging component in order to log the first event;

wherein, in response to said first client-side logging component calling the server-side logging component in order to log the first event, the server-side logging component is executable to utilize the means for persistently storing information in order to persistently log the first event.

20. The system of claim 19, further comprising:

a third computer system including a CPU and memory;

a second client-side logging component comprised in the memory of the third computer system;

a second executable module running on the third computer system, wherein the second executable module is executable to call the second client-side logging component in order to log a second event;

wherein, in response to said second executable module calling the second client-side logging component in order to log a second event, the second client-side logging component is executable to call the server-side logging component in order to log the second event;

wherein, in response to said second client-side logging component calling the server-side logging component in order to log the second event, the server-side logging component is executable to utilize the means for persistently storing information in order to persistently log the second event.

21. The system of claim 20;

wherein the first client-side logging component executes in-process with the first executable module;

wherein the second client-side logging component executes in-process with the second executable module.

22. A method for logging an event, comprising:

a first executable module calling a first client-side logging component in order to log an event, wherein the first executable module and the first client-side logging component are stored in the memory of a computer system and execute on the computer system;

the first client-side logging component calling a server-side logging component in order to log the event, in response to said first executable module calling the first client-side

logging component, wherein the server-side logging component is stored in the memory of the computer system and executes on the computer system;

wherein the event comprises an event level;

wherein said first executable module calling the first client-side logging component comprises the first executable module passing information specifying the event level to the first client-side logging component;

wherein said first client-side logging component calling the server-side logging component comprises the first client-side logging component passing information specifying the event level to the server-side logging component;

the server-side logging component persistently storing information in order to persistently log the event, in response to said first client-side logging component calling the server-side logging component.;

wherein said server-side logging component persistently logging the event comprises the server-side logging component persistently logging the event level.

23. The method of claim 22,

wherein the first client-side logging component maintains logging criteria information specifying which levels of events should be logged;

wherein, in response to said first executable module calling the first client-side logging component, the first client-side logging component compares the information specifying the event level to the logging criteria information, in order to determine whether the event should be logged;

wherein, if the first client-side logging component determines that the event should not be logged, the client-side logging component does not perform said calling the server-side logging component.

24. The method of claim 22,

wherein the event is associated with a first event category;

wherein said first executable module calling the first client-side logging component comprises the first executable module passing information specifying the first event category to the first client-side logging component;

wherein said first client-side logging component calling the server-side logging



component comprises. the first client-side logging component passing information specifying the first event category to the server-side logging component;

wherein said server-side logging component persistently logging the event comprises the server-side logging component persistently logging the first event category.

25. The method of claim 24,

wherein the first client-side logging component maintains logging criteria information specifying which categories of events should be logged;

wherein, in response to said first executable module calling the first client-side logging component, the first client-side logging component compares the information specifying the first event category to the logging criteria information, in order to determine whether the event should be logged;

wherein, if the first client-side logging component determines that the event should not be logged, the client-side logging component does not perform said calling the server-side logging component.

26. The method of claim 25,

wherein the event is also associated with a second event category;

wherein said first executable module calling the first client-side logging component further comprises the first executable module passing information specifying the second event category to the first client-side logging component;

wherein, in response to said first executable module calling the first client-side logging component, the first client-side logging component compares the information specifying the second event category to the logging criteria information, in order to determine whether the event should be logged.

27. The method of claim 22,

wherein the first client-side logging component executes in-process with the first executable module.

28. The method of claim 22, further comprising:

a second executable module calling the first client-side logging component in order to log an event, wherein the second executable module is stored in the memory of the

computer system and executes on the computer system;

wherein the first executable module and the second executable module and the first client-side logging component run in the same process.

29. The method of claim 28,  
wherein the first client-side logging component executes as an in-process COM object.

30. The method of claim 22, further comprising:  
a second executable module calling a second client-side logging component in order to log an event, wherein the second executable module and the second client-side logging component are stored in the memory of the computer system and execute on the computer system;

wherein the first executable module and the first client-side logging component run in a first process;

wherein the second executable module and the second client-side logging component run in a second process.

31. A method for logging an event, comprising:  
a first executable module calling a first client-side logging component in order to log an event, wherein the first executable module and the first client-side logging component are stored in the memory of a computer system and execute on the computer system;

the first client-side logging component calling a server-side logging component in order to log the event, in response to said first executable module calling the first client-side logging component, wherein the server-side logging component is stored in the memory of the computer system and executes on the computer system;

wherein, in response to said first executable module calling the first client-side logging component in order to log an event, the first client-side logging component stores the event in a queue and return execution control to the first executable module prior to evaluating the event;

wherein said first client-side logging component calling the server-side logging component comprises the first client-side logging component asynchronously retrieving the event from the queue and then calling the server-side logging component;

the server-side logging component persistently storing information in order to persistently log the event, in response to said first client-side logging component calling the server-side logging component.

32. The method of claim 31,

wherein said first client-side logging component asynchronously retrieving the event from the queue and then calling the server-side logging component is performed by an event queue manager thread.

33. The method of claim 22,

a logging administration tool configuring the server-side logging component with event-logging criteria information, wherein the event-logging criteria information includes information specifying which levels of events should be logged;

the server-side logging component propagating the event-logging criteria information to the first client-side logging component in response to said logging administration tool configuring the server-side logging component with event-logging criteria information;

wherein the first client-side logging component maintains logging criteria information specifying which levels of events should be logged;

wherein the first client-side logging component updates the logging criteria information maintained by the first client-side logging component in response to said server-side logging component propagating the event-logging criteria information to the first client-side logging component.

34. The method of claim 33,

wherein the server-side logging component is a COM component;

wherein the first client-side logging component is a COM component;

wherein said server-side logging component propagating the event-logging criteria information to the first client-side logging component is performed by using COM mechanisms for sending a COM event from one COM component to another COM component.

35. The method of claim 22, further comprising:

a logging administration tool configuring the server-side logging component with event-logging criteria information, wherein the event-logging criteria information includes

information specifying which categories of events should be logged;

the server-side logging component propagating the event-logging criteria information to the first client-side logging component in response to said logging administration tool configuring the server-side logging component with event-logging criteria information;

wherein the first client-side logging component maintains logging criteria information specifying which categories of events should be logged;

wherein the first client-side logging component updates the logging criteria information maintained by the first client-side logging component in response to said server-side logging component propagating the event-logging criteria information to the first client-side logging component.

36. The method of claim 22,

wherein, in response to said first executable module calling the first client-side logging component in order to log an event, the first client-side logging component is executable to timestamp the event;

wherein said client-side logging component calling the server-side logging component comprises the client-side logging component passing timestamp information for the event to the server-side logging component;

wherein said server-side logging component persistently storing information in order to persistently log the event comprises the server-side logging component logging the timestamp information.

37. The method of claim 36,

wherein the server-side logging component is executable to utilize event timestamp information in order to determine the order in which events should be logged.

38. The method of claim 22,

wherein said server-side logging component persistently storing information in order to persistently log the event comprises the server-side logging component logging the information into an element from the group consisting of:

a file and a database.

39. A method for logging an event, comprising:

a first executable module calling a first client-side logging component in order to log a first event, wherein the first executable module and the first client-side logging component are stored in the memory of a first computer system and execute on the first computer system;

the first client-side logging component calling a server-side logging component in order to log the first event, in response to said first executable module calling the first client-side logging component, wherein the server-side logging component is stored in the memory of a second computer system and executes on the second computer system;

the server-side logging component persistently storing information in order to persistently log the first event, in response to said first client-side logging component calling the server-side logging component.

40. The method of claim 39, further comprising:

a second executable module calling a second client-side logging component in order to log a second event, wherein the second executable module and the second client-side logging component are stored in the memory of a third computer system and execute on the third computer system;

the second client-side logging component calling the server-side logging component in order to log the second event, in response to said second executable module calling the second client-side logging component,

the server-side logging component persistently storing information in order to persistently log the second event, in response to said second client-side logging component calling the server-side logging component.

41. The method of claim 40,

wherein the first client-side logging component executes in-process with the first executable module;

wherein the second client-side logging component executes in-process with the second executable module.